arXiv:2304.09664v1 [cs.DM] 19 Apr 2023

# 关于块并行自动机网络的计数与枚举

Kévin Perrot[1,2], Sylvain Sené[1,2], 和 Léah Tapin[2]

[1]公立大学，马赛，法国[2]Aix-Marseille Univ, CNRS，LIS，马赛，法国

**摘要**

当我们从可计算性/复杂性和建模两个角度研究有限动力系统时，自动机网络似乎是一种特别适合用于开展理论研究的数学模型。在本文中，自动机网络是指由有限个实体（即自动机）组成的集合，每个自动机都有其自身的一组可能状态，这些自动机在离散时间内相互作用，而这种相互作用由局部函数定义，使自动机能够根据其邻域的状态改变自身状态。关于这一计算模型的研究强调了自动机更新其状态的方式（即 调度机制）的重要性，也就是更新模式，这些模式可以是确定性的、周期性的、公平的，也可以不具备这些性质。事实上，给定的网络可能具有多种底层动力学，而这些动力学在很大程度上取决于我们允许该网络在何种更新模式下演化。本文关注的是一种近期在建模框架中引入的新型更新模式族，该更新模式族具有确定性、周期性和公平性，其名称为块并行更新模式，以此与广为人知且已有深入研究的块顺序更新模式相对应。更具体地说，在自动机网络的一般背景下，本研究旨在阐明块并行更新模式与块顺序更新模式之间的区别，并对其进行计数和枚举：一方面，从绝对意义上进行计数和枚举，仅保留能够产生不同动力学的代表；另一方面，仅保留能够产生不同同构极限动力学的代表。总而言之，本文构成了对这些更新模式及其对自动机网络动力学影响的首次理论分析。

## 1 引言

自动机网络诞生于现代计算机科学的开端，即1940年代，尤其源于McCulloch 和 Pitts [26] 关于神经网络的奠基性工作，以及 von Neumann关于元胞自动机的研究 [39], ；元胞自动机自那以后已成为得到广泛研究的计算模型。前者通常被划分为有限且异质的结构（图），而后者则被划分为无限但规则的结构（晶格）。尽管二者之间存在深刻差异，但它们都属于自动机网络这一类模型；该类模型汇集了所有这样的模型：它们通过自动机在离散时间内相互作用，在局部层面进行定义，从而使其运行的全局计算从支配它们的这些局部交互中涌现出来。这些初始模型引发了围绕可计算性理论 [25, 11, 36, 4, 5] 和复杂度 [21, 13]的大量研究。

除了这些对理论计算机科学的贡献外，20世纪60年代末凸显了有限自动机网络的重要作用，而本文正是聚焦于此，

尤其是布尔自动机网络在生物网络定性建模中的重要作用，这要归功于 Kauffman [23]和 Thomas [38], 的杰出研究。他们率先强调，基因表达谱可以通过此类模型，以及其底层动力系统所呈现的极限行为来刻画；这些极限行为例如可以代表表型、细胞类型，甚至生物节律。此后，（布尔）自动机网络及其扩展形式便成为基因调控定性建模中应用最广泛的离散模型 [37, 27, 35, 7, 22]。

从基础角度来看，给定一个自动机网络 $f$ 以及两种不同的方式来随时间更新其自动机的状态，即 两种不同的更新模式$\mu$ 和 $\mu'$，其对应的两个动力系统 $f_\mu$ 和 $f_{\mu'}$ 显然可能不同。换言之，更新模式对自动机网络的动力学起着至关重要的作用，自 Robert 关于离散迭代的开创性研究 [33, 34], 以来，深入理解其影响已成为该领域的热点，并在过去二十年中促成了大量进一步研究 [15, 2, 14, 28, 3, 30, 32, 31]。这一课题从理论和应用角度来看都尤为重要。事实上，*(i)* 更新模式可能具有不同的性质（确定性的或非确定性的、周期性的或非周期性的，.....），其数量是无限且不可数的；并且 *(ii)* 例如，如果我们将自动机网络视为基因调控网络的模型，那么即使染色质动力学似乎发挥着关键作用 [16, 12], 我们仍然不知道哪些“自然调度”支配着基因表达和调控。

在 [9], 生物调控建模框架下，作者提出了一种新的周期性更新模式族，并着重强调其“自然”的计算能力：这一模式族具有能够打破不动点集不变这一经典性质的特征（一个周期内可以出现局部更新重复），并且能够捕捉遗传或生理性质/起源的内源性生物计时器/时钟，例如染色质动力学所诱导的计时器/时钟。在此之前，探讨周期更新模式作用的研究聚焦于块顺序更新模式，即将自动机划分为一个子集列表，使同一子集中的自动机同时更新其状态，而各子集则依次迭代。块并行更新模式则以对偶方式定义。它们不是定义为自动机集合的有序划分，而是定义为列表集，或称“分区序”，使得同一列表中的自动机按照该列表的周期依次更新其状态，而各列表则在初始时间步同时启动。

由于局部更新重复能够生成非预期的动力系统极限集，并且它们在建模方面似乎具有良好的应用前景，我们确信有必要在不同的框架下对其进行详细研究。本文首次从理论上分析了自动机网络所依托的离散动力系统背景下，与块并行更新模式相关的组合方面。这种对更新模式本身的形式化引入，将为块并行自动机网络的动力学进一步发展奠定坚实基础，并有望为后续研究铺平道路。与经典的块顺序更新模式相比，块并行自动机网络的动力学要难以理解得多。

第2节介绍了全文中使用的主要定义和符号。第3节阐述了我们的主要贡献，并分为五个部分。第3.1节讨论了块顺序与块并行更新模式之间的交集；特别是，它刻画了块顺序模式何时属于块并行模式，反之亦然。第3.2节旨在从绝对意义上研究块并行模式；我们给出了两个简洁的闭式公式来对其进行计数（其中一个源自已有文献），并提供了一种枚举它们的算法。在完成数值模拟之后，

在研究块并行自动机网络的动力学时，我们观察到，对于任意自动机网络，某些块并行模式本质上是相似的，因为它们总是产生完全相同的动力学。第3.3节给出了按照自动机网络动力学计数本质不同的块并行模式的闭式公式，并开发了一种枚举算法。第3.4节则给出了本质不同的块并行模式的闭式公式和枚举算法，但此处是针对自动机网络的同构极限动力学。正如第3.5节展示的数值实验所表明的，对于关注动力系统渐近（极限）行为的情况，这能将需要考察的元素数量减少若干个数量级。我们的代码可在网上免费获取（见第3.5节）。最后，我们在第4节展望了本研究的后续方向。

## 2 定义

令 $[\![n]\!] = \{0, \ldots, n-1\}$，令 $\mathbb{B} = \{0, 1\}$，令 $x_i$ 表示向量 $x \in \mathbb{B}^n$ 的第 $i$ 个分量，令 $x_I$ 表示对于自动机 $I \subseteq [\![n]\!]$ 的某个子集，将 $x$ 投影到 $\mathbb{B}^{|I|}$ 的一个元素上，令 $e_i$ 为第 $i$ 个基向量，并且 $\forall x,\ y \in \mathbb{B}^n$，令 $x+y$ 表示模二按位加法。令 $\sigma^i$ 表示数列上阶数为 $i \in \mathbb{Z}$ 的循环的移位（将位置 0 处的元素移向位置 $i$）。令 $\sim$ 表示图同构，即 对于 $G = (V,\ A)$ 和 $G' = (V',\ A')$，当且仅当存在一个双射 $\pi\colon V \to V'$，使得 $(u,\ v) \in A \iff (\pi(u)\ \pi(v)) \in A'$。

**自动机网络** 一个大小为 $n$ 的自动机网络 (AN)是一个离散动力系统，由一组 $n$ 自动机 $[\![n]\!]$ 组成，每个自动机在有限字母表 $X_i$ 中持有一个状态$i \in [\![n]\!]$。构型 是 $X = \prod_{i \in [\![n]\!]} X_i$的一个元素。AN由函数$f\colon X \to X$定义，该函数分解为 $n$ 局部函数 $f_i\colon X \to X_i$，其中 $f_i$ 是 $f$的第$i$个分量。为了使系统演化，必须定义自动机何时使用其局部函数更新状态，这可以通过多种方式实现。

**块顺序更新模式** 一个数列 $(W_\ell)^{\ell \in [\![p]\!]}$，其中 $W_\ell \subseteq [\![n]\!]$ 对所有 $\ell \in [\![p]\!]$均成立，当且仅当它是一个有序划分：

$$\bigcup_{\ell \in [\![p]\!]} W_\ell = [\![n]\!] \text{ and } \forall i, j \in [\![p]\!], i \neq j \implies W_i \cap W_j = \emptyset.$$

当更新模式 $\mu = (W_\ell)^{\ell \in [\![p]\!]}$ 是块顺序 时，称其为块顺序更新模式，其中 $\mu$ 是一个有序划分，而 $W_\ell$ 称为块。大小为 $n$ 的块顺序更新模式集合记为 $\mathrm{BS}_n$。在 $\mu \in \mathrm{BS}_n$ 下对 $f$ 的更新由 $f_{(\mu)}$ 给出，具体如下：

$$f_{(\mu)}(x) = f_{(W_{p-1})} \circ \cdots \circ f_{(W_1)} \circ f_{(W_0)}(x),$$

其中对于所有 $\ell \in [\![p]\!]$:

$$\forall i \in [\![n]\!], f_{(W_\ell)}(x)_i = \begin{cases} f_i(x) & \text{if } i \in W_\ell, \\ x_i & \text{otherwise.} \end{cases}$$

**块并行更新模式** 在块顺序更新模式中，一个块内的自动机同时更新，而各块则依次更新。块并行更新模式基于对偶原理：一个块内的自动机依次更新，而各块则同时更新。它不是被定义为一个

无序块序列，因此块并行更新模式将被定义为有序块的集合。集合 $\{S_k\}_{k\in[\![s]\!]}$ 含有 $S_k = (i_0^k, \dots, i_{n_k-1}^k)$ 个元素的数列，对于所有 $k \in [\![s]\!]$ 是分区序，当且仅当：

$$\bigcup_{k\in[\![s]\!]} S_k = [\![n]\!] \text{ and } \forall i,j \in [\![s]\!], i \neq j \implies S_i \cap S_j = \emptyset.$$

当 $\mu$ 是分区序时，更新模式 $\mu = \{S_k\}_{k\in[\![s]\!]}$ 被称为 块并行，而数列 $S_k$ 被称为 *o*-块（即 有序块）。大小为 $n$ 的块并行更新模式集合记为 $\mathrm{BP}_n$。令 $p =$ 为 $\mathrm{lcm}(n_1, \dots, n_s)$，则在 $\mu \in \mathrm{BP}_n$ 下对 $f$ 的更新定义为 $f_{\{\mu\}}$： $X \to X$ 如下：

$$f_{\{\mu\}}(x) = f_{(W_{p-1})} \circ \cdots \circ f_{(W_1)} \circ f_{(W_0)}(x),$$

其中，对所有 $\ell \in [\![p]\!]$，我们定义 $W_\ell = \{i_{\ell \bmod n_k}^k \mid k \in [\![s]\!]\}$。

**基本考虑**正如上述定义所自然表明的那样，可以将具有 $S_k = (i_0^k, \dots, i_{n_k-1}^k)$ 的块并行更新模式 $\{S_k\}_{k\in[\![s]\!]}$ 转换为长度为 $p = \mathrm{lcm}(n_1, \dots, n_s)$ 的块序列。我们将其定义为 $\varphi$：

$$\varphi(\{S_k\}_{k\in[\![s]\!]}) = (W_\ell)_{\ell\in[\![p]\!]} \text{ with } W_\ell = \{i_{\ell \bmod n_k}^k \mid k \in [\![s]\!]\}.$$

为了区分块序列和 o-块集合，我们将由 $f$ 在 $\mu$ 为块序列（分别为 o-块集合）时所诱导的动力系统记为$f_{(\mu)}$（分别为 $f_{\{\mu\}}$），在上下文明确时则简记为 $f_\mu$。此外，滥用记号，我们将 $[\![n]\!]$ 作为块序列时的分区序集合记为 $\varphi$（$\mathrm{BP}_n$）。

块顺序和块并行更新模式具有周期性（每一步都重复相同的更新过程），并且具有公平性（每个自动机在每一步中至少更新一次）。我们区分步和子步骤的概念。一步是 $x$ 与 $f_{(\mu)}(x)$（或 $f_{\{\mu\}}(x)$）之间的时间区间，可以划分为 $p = |\mu|$（或 $p = |\varphi(\mu)| = \mathrm{lcm}(n_1, \dots, n_s)$）个子步骤，对应于每次仅更新一个自动机块的基本时间区间。最基本的更新模式是并行 $\mu_{\mathsf{par}}$，它在每一步同时更新所有自动机。它是元素 $([\![n]\!]) \in \mathrm{BS}_n$ 和 $\{(i) \mid i \in [\![n]\!]\} \in \mathrm{BP}_n$，其中$\varphi(\{(i) \mid i \in [\![n]\!]\}) = ([\![n]\!])$。

**注1.** 请注意，在块顺序更新模式下，每个自动机在一步中恰好更新一次；而在块并行更新模式下，有些自动机在一步中可能被更新多次。重复更新可能对极限动力学产生诸多影响。例如，$n = 3$ 自动机网络，其中 $f_i(x) = x_{i-1 \bmod n}$(即 一个大小为 3 的正环)，在更新模式 $\mu = (\{1,2\}, \{0,2\}, \{0,1\})$ 下有 4 个不动点，其中 2 个无法通过块顺序更新模式得到（在此例中为 $\mu \notin \mathrm{BP}_n$）。

**注2.** 设 $\mu = \{S_k\}_{k\in[\![s]\!]}$ 是一种块并行更新模式。$\varphi(\mu)$ 的每个块大小相同，即 $s$；此外，$\varphi(\mu)$ 的每个块都是唯一的。

**不动点、极限环和吸引子**令 $f_\mu$ 为由大小为 $n$ 的自动机网络 $f$ 和更新模式 $\mu$ 定义的动力系统。

令 $p \geq 1$。构型数列 $x^0$、……、$x^{p-1} \in X$ 是 $f_\mu$ 的极限环 当且仅当 $\forall i \in [\![p]\!]$，$f_\mu(x^i) = x^{i+1 \bmod p}$。长度为 $p = 1$ 的极限环是不动点。构型数列 $x^0$，$x^1$、……、$x^{p-1} \in X$ 是吸引子 当且仅当它是一个极限环，并且存在 $x \in X$ 和 $i \in [\![p]\!]$，使得 $f_\mu(x) = x^i$，但 $x \notin \{x^0, \dots, x^{p-1}\}$。

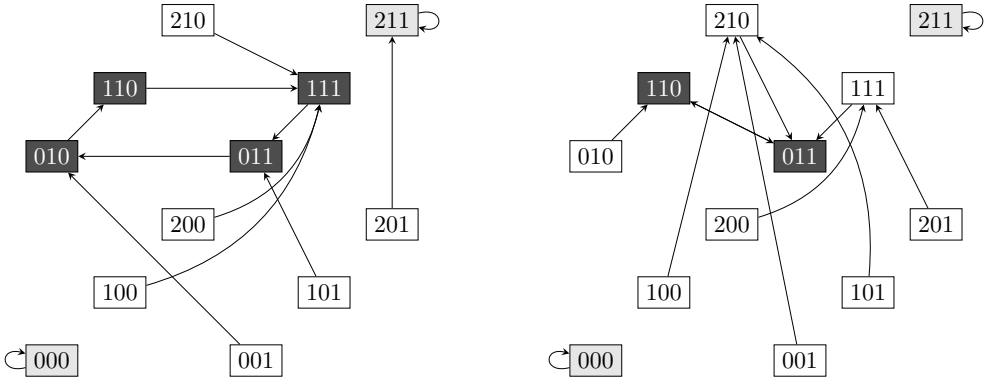


图1：来自例1的$f_{(\mu_{\mathsf{bs}})}$（左）和 $f_{\{\mu_{\mathsf{bp}}\}}$（右）的动力学。

**例1**。令 $f\colon [\![3]\!]\times\mathbb{B}\times\mathbb{B}\to[\![3]\!]\times\mathbb{B}\times\mathbb{B}$ 为定义如下的自动机网络：

$$f(x)=\begin{pmatrix} f_0(x) & = \begin{cases} 0 & \text{if } ((x_0=0)\wedge(x_1=x_2))\vee(x_0=x_1=x_2=1) \\ 1 & \text{if } x_1+x_2 \mod 2=1 \\ 2 & \text{otherwise} \end{cases} \\ f_1(x) & =(x_0\neq 0)\vee x_1\vee x_2 \\ f_2(x) & =((x_0=1)\wedge x_1)\vee(x_0=2) \end{pmatrix}.$$

设 $\mu_{\mathsf{bs}}=(\{1\},\ \{0,2\})$ 和 $\mu_{\mathsf{bp}}=\{(0),\ (2,\ 1)\}$。更新模式 $\mu_{\mathsf{bs}}$ 是块顺序的，而 $\mu_{\mathsf{bp}}$ 是块并行的，其中 $\varphi(\mu_{\mathsf{bp}})=(\{0,2\},\ \{0,1\})$。系统 $f_{(\mu_{\mathsf{bs}})}$ 和 $f_{\{\mu_{\mathsf{bp}}\}}$ 的动力学不同，如图1所示。它们都有相同的两个不动点和一个极限环，但相似之处仅此而已。$f_{(\mu_{\mathsf{bs}})}$ 的极限环大小为 4，而 $f_{\{\mu_{\mathsf{bp}}\}}$ 的极限环大小为 2。此外，$f_{\{\mu_{\mathsf{bp}}\}}$ 的两个不动点都不是吸引子，而 $f_{(\mu_{\mathsf{bs}})}$ 的一个不动点，即 211，是吸引子。这两种更新模式的动力学在 $\mathsf{BP}_3\cup\mathsf{BS}_3$ 中都是独一无二的。

## 3 块并行更新模式的计数与枚举

本节其余部分中，令 $p(n)$ 表示 $n$ 的整数分拆数（和为 $n$ 的整数多重集），令 $d(i)$ 表示 $n$ 的第 $i$ 个分拆中的最大部分大小，令 $m(i,j)$ 表示 $n$ 的第 $i$ 个分拆中大小为 $j$ 的部分的重数。举例来说，令 $n=31$，并假设第 $i$ 个分拆为 (2, 2, 3, 3, 3, 3, 5, 5, 5)，则有 $d(i)=5$，且 $m(i,1)=0$，$m(i,2)=2$，$m(i,3)=4$，$m(i,4)=0$，$m(i,5)=3$。一个划分将作为分区序的支撑集，其中每个部分都是一个 o-块。在我们的例子中，可以有：

$$\{(0,1),(2,3),(4,5,6),(7,8,9),(10,11,12),(13,14,15),\\ (16,17,18,19,20),(21,22,23,24,25),(26,27,28,29,30)\},$$

我们将其描绘为如下的矩阵表示：

$$\begin{pmatrix}0&1\\2&3\end{pmatrix}\begin{pmatrix}4&5&6\\7&8&9\\10&11&12\\13&14&15\end{pmatrix}\begin{pmatrix}16&17&18&19&20\\21&22&23&24&25\\26&27&28&29&30\end{pmatrix}.$$

我们将大小为 $j\cdot m(i,\ j)$ 的元素称为矩阵，并将它们记作 $M_1$、.....、$M_{d(i)}$，其中 $M_j$ 有 $m(i,\ j)$ 行和 $j$ 列（当 $m(i,\ j)=0$ 时，$M_j$ 为空）。该划分定义了矩阵的维度，且每一行都是一个 o- 块。

作为比较，块顺序更新模式（$[\![n]\!]$的有序划分）由 OEIS 的数列 A000670 的有序贝尔数给出 [18, 29]。其闭式公式为：

$$|\mathsf{BS}_n| = \sum_{i=1}^{p(n)} \frac{n!}{\prod_{j=1}^{d(i)} (j!)^{m(i,j)}} \cdot \frac{\left(\sum_{j=1}^{d(i)} m(i,j)\right)!}{\prod_{j=1}^{d(i)} m(i,j)!}.$$

直观而言，$n$ 的一个有序划分为构造块顺序更新模式提供了一种依据：将 $[\![n]\!]$ 中的元素置于各个块中，并考虑块内的置换。这是左侧的分数：$n!$ 除以每个大小为 $j$的块所对应的 $j!$，同时考虑重数。右侧的分数则用于修正计数，因为我们对 $p(n)$求和时，考虑的是 $n$ 的（无序）划分：$n$ 的每个划分都可能通过对所有块进行排序（分子，其中重数之和即为块的数量），产生 $n$的不同有序划分；而对于大小相同的块，由于它们之间的置换不会产生影响，因此需要在分母中予以消除。前十项为（从$n = 1$ 开始）：

$$1, 3, 13, 75, 541, 4683, 47293, 545835, 7087261, 102247563.$$

### 3.1 块顺序与块并行模式的交集

为了能够比较块顺序与块并行更新模式，本文将二者都以块序列形式书写（块顺序更新模式的经典形式，以及块并行更新模式的重写形式）。

首先，我们知道 $\varphi(\mathsf{BP}_n) \cap \mathsf{BS}_n$ 非空，因为它至少包含

$$\mu_{\mathsf{par}} = ([\![n]\!]) = \varphi(\{(0), (1), \ldots, (n-1)\}).$$

然而，$\mathsf{BS}_n \subseteq \varphi(\mathsf{BP}_n)$ 和 $\varphi(\mathsf{BP}_n) \subseteq \mathsf{BS}_n$ 都不成立。实际上，$\mu_s = (\{0,1\},\{2\}) \in \mathsf{BS}_3$，但 $\mu_s \notin \varphi(\mathsf{BP}_3)$，因为块并行自动机在其顺序形式中不能包含大小不同的块。类似地，$\mu_p = \varphi(\{(1,2),(0)\}) = (\{0,1\},\{0,2\}) \in \mathsf{BP}_3$，但 $\mu_p \notin \mathsf{BS}_3$，因为自动机 0 被更新了两次。尽管如此，我们仍然可以精确定义 $\mathsf{BS}_n \cap \varphi(\mathsf{BP}_n)$ 的交集。

**引理 1.** 设 $\mu$ 是一种更新模式，它被写成 $[\![n]\!]$ 中元素的块序列。那么 $\mu \in (\mathsf{BS}_n \cap \varphi(\mathsf{BP}_n))$ 当且仅当 $\mu$ 是一个有序划分，且 $\mu$ 的所有块大小都相同。

证明。令 $n \in \mathbb{N}$。

($\Longrightarrow$) 令 $\mu \in (\mathsf{BS}_n \cap \varphi(\mathsf{BP}_n))$。由于 $\mu \in \mathsf{BS}_n$，$\mu$是一个有序划分。此外，$\mu \in \varphi(\mathsf{BP}_n)$，因此所有 $\mu$的块大小相同（备注 2）。

($\Longleftarrow$) 令 $\mu = (W_\ell)^{\ell \in [\![p]\!]}$ 为 $[\![n]\!]$ 的一个有序划分，其所有块的大小均相同，记为 $s$。由于 $\mu$ 是一个有序划分，$\mu \in \mathsf{BS}_n$。对于每个 $\ell \in [\![p]\!]$，我们可以任意地将 $W_\ell$ 中的元素从 0 到 $s-1$ 编号为 $W_\ell = \{W_\ell^0, \ldots, W_\ell^{s-1}\}$。现在，我们按如下方式定义数列集合 $\{S_k\}_{k \in [\![s]\!]}$：$\forall k \in [\![s]\!]$，$S_k = \{W_\ell^k \mid \ell \in [\![p]\!]\}$。这是一个分区序，使得 $\varphi(\{S_k\}_{k \in [\![s]\!]}) = \mu$，这意味着 $\mu \in \varphi(\mathsf{BP}_n)$。 □

**推论 1**。如果 $\mu \in \mathsf{BP}_n$ 由大小为 $p$ 的 $s$ *o*-块组成，那么 $\varphi(\mu) \in \mathsf{BS}_n$ 由大小为 $s$ 的 $p$ 块组成。

由引理 1 和推论 1 可知，给定 $n \in \mathbb{N}$，$\mathsf{SEQ}_n$ 中的顺序更新模式集合满足：每个自动机在每个步中恰好更新一次，且每个子步中仅更新一个自动机；该集合是 $(\mathsf{BS}_n \cap \varphi(\mathsf{BP}_n))$ 的子集。

此外，我们可以陈述以下命题，用以计算同时属于 $\mathsf{BS}_n$ 和 $\varphi(\mathsf{BP}_n)$ 的块序列数量。

**命题 1.** 给定 $n \in \mathbb{N}$，我们有：

$$|\mathsf{BS}_n \cap \varphi(\mathsf{BP}_n)| = \sum_{d|n} \frac{n!}{(\frac{n}{d}!)^d}.$$

证明。 证明直接源自 OEIS 的数列 A061095 [20], ，该数列统计将 $n$ 带标签的元素划分到带标签的盒子中，且每个盒子中的元素数量相同的方式。在我们的语境中，“元素”是自动机，而“带标签的盒子”是有序划分中的块。 □

### 3.2 分区序

块并行更新模式可表示为分区序，即一个（无序的）（有序）数列集合。OEIS 将这一概念记录为数列 A000262 [19], ，并将其描述为“列表集”的数量。它有一个简洁的闭式公式：

$$|\mathsf{BP}_n| = \sum_{i=1}^{p(n)} \frac{n!}{\prod_{j=1}^{d(i)} m(i,j)!}.$$

直观而言，对于每个分拆，填满所有矩阵（将 $[\![n]\!]$ 个元素放置到 $n$ 个位置的方式）后，再对每个矩阵中的行进行置换（矩阵 $M_j$ 有 $m(i,\ j)$ 行）。命题 2 给出了另一个闭式公式。该公式对于生成所有块并行更新模式尤其有用，因为其中的各部分有助于我们构造算法 1，以枚举 $\mathsf{BP}_n$ 的分区序。

**命题 2.** 对于任意 $n \geq 1$，我们有：

$$|\mathsf{BP}_n| = \sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,k)}{j \cdot m(i,j)} \cdot \frac{(j \cdot m(i,j))!}{m(i,j)!}.$$

证明。 每个划分都可作为生成不同分区序的一个支撑（对 $i$ 求和）；具体而言，对每个矩阵（对 $j$ 求积），考虑从其包含的 $[\![n]\!]$ 个元素中选取 $j \cdot m(i,\ j)$ 个元素的所有组合（从剩余元素中选取的二项式系数），以及将这些元素排序的所有方式，并对行的置换取商（阶乘之比）。注意，将二项式系数用 $\binom{x}{y} = {}_{y\,.}{}^{x!}{}_{!(x-y)!}$ 展开可得

$$\prod_{j=1}^{d(i)} \binom{n - \sum_k}{j \cdot m(i,j)} \cdot (j \cdot m(i,j))! = \prod_{j=1}^{d(i)} \frac{(n - \sum_k)!}{(n - \sum_k - j \cdot m(i,j))!} = \frac{n!}{0!} = n!,$$

whe 其中 $\sum_k$ 是 $\sum_{k=1}^{j-1} k \cdot m(i,\ k)$ 的简写，由此可得 OEIS 公式。 □

前十项（从 $n = 1$ 开始）为：

$$1, 3, 13, 73, 501, 4051, 37633, 394353, 4596553, 58941091.$$

算法 1 枚举大小为 $n$ 的分区序。对于 $n$ 的每个划分 $i$（第 2 行，可按照 [24], 枚举整数 $n$ 的划分，以便计算 $d(i)$ 和 $m(i,j)$），它会按照本节引言中的说明，将所有块并行更新模式枚举为一个矩阵列表。矩阵逐个填充（辅助函数的参数$j$ ），并使用递归（当一个矩阵准备就绪时，它会调用 on

**算法 1：** BP的枚举$n$

```
1  函数EnumBP(n)：
2      对每个 i 从 1 到 p(n)执行
3          EnumBPaux(n, i, 1)

4  函数 EnumBPaux(n, i, j, M_1, ..., M_{j-1})：
5      如果 d(i) < j 则
6          枚举(M)
7          返回
8      如果 m(i,j) > 0 则
9          对 大小为 j·m(i,j) 的组合 A，在 [[n]] \ ⋃_{k=1}^{j-1} M_k 中执行
10             对 大小为 m(i,j) 的组合 C，在 A 中执行
11                 排序(C)
12                 for k ← 1 to m(i,j) 执行
13                     M_j[k][1] ← C[k]
14                 foreach 置换 B of A \ C do
15                     对于 k ← 1 至 m(i,j) 执行
16                         for ℓ ← 2 to j do
17                             M_j[k][ℓ] ← B[(k-1)·(j-1)+ℓ-1]
18                     EnumBPaux(n, i, j+1, M_1, ..., M_{j-1}, M_j)
19     else
20         EnumBPaux(n,i,j+1, M_1,...,M_{j-1},∅)
```

$j+1$ 第18行或第20行）。当 $d(i) < j$时，块并行更新模式已完成枚举（第6行）。当 $m(i,j) = 0$时，矩阵 $M_j$ 为空，且会立即进行递归调用（第20行）。否则，我们选择 $A$ 个 $j \cdot m(i,j)$ 元素组成的集合，从 $[\![n]\!]$ 中取出并放入矩阵 $M_j$，即 在 $\binom{n}{j \cdot m\, i(,j)}$ 种可能的组合中选择一种（第9行）。为了枚举所有行置换意义下的这类矩阵，我们选择第一列的元素（第10行），并按预定义的顺序放置它们（第11—13行）。矩阵的其余部分以所有可能的方式填充（第14—17行），并针对每种可能执行一次递归调用（第18行），以便对剩余矩阵及 $[\![n]\!]$的剩余元素重复此过程。

**定理 1**。 算法 *1* 正确：它枚举了$\mathsf{BP}_n$。

证明。 首先，它为 $n$ 枚举了正确数量的更新模式。事实上，对于 $n$ 的每个划分，以及从 1 到 $d(i)$ 的每个 $j$，我们选择 $j \cdot m(i,\ j)$ 个元素来填充矩阵$M_j$。然后，我们从中选择第一列的 $m(i,\ j)$ 个元素，并对该矩阵中的其余元素进行所有置换。由此得到以下方程：

$$
\begin{aligned}
|\texttt{EnumBP}(n)| &= \sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,j)}{j \cdot m(i,j)} \binom{m(i,j) \cdot j}{m(i,j)} (m(i,j)(j-1))! \\
&= \sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,j)}{j \cdot m(i,j)} \frac{m(i,j) \cdot j!}{(m(i,j))!} \\
&= |\mathsf{BP}_n|.
\end{aligned}
$$

其次，由于每个矩阵的第一列始终按升序排列，该算法无法枚举出两个仅通过行的置换即可相同的矩阵表示。因此，它不会两次返回同一个分区序，这意味着它确实枚举了每一个分区序。

□

### 3.3 动力学相等下的划分序

对于块顺序更新模式，给定一个自动机网络 $f$ 以及两个块并行更新模式 $\mu$ 和 $\mu'$，$f$ 在 $\mu$ 下的动力学可能与 $f$ 在 $\mu'$ 下的动力学相同。进一步来说，在块并行更新模式的框架下，存在一些更新模式对 $\mu, \mu'$，使得对于任意自动机网络 $f$，动力学 $f_{\{\mu\}}$ 与 $f_{\{\mu'\}}$ 完全相同。因此，为了对所有可能的动力学进行穷举搜索，无须将它们全部生成。我们用以下等价关系将其形式化。

**定义 1.** 对于 $\mu, \mu' \in \mathsf{BP}_n$，我们记作 $\mu \equiv_0 \mu'$ 当 $\varphi(\mu) = \varphi(\mu')$。

以下引理表明，这一等价关系对于大小为 $n$ 的 AN 是必要且充分的。

**引理 2。** 对于任意 $\mu$、$\mu' \in \mathsf{BP}_n$，有 $\mu \equiv_0 \mu' \iff \forall f\colon\ X \to X$、$f_{\{\mu\}} = f_{\{\mu'\}}$.

证明。设 $\mu$ 和 $\mu'$ 是 $\mathsf{BP}_n$ 的两种块并行更新模式。

($\Longrightarrow$) 考虑 $\mu \equiv_0 \mu'$，并令 $f\colon\ X \to X$ 为一个自动机网络。于是，我们有 $f_{\{\mu\}} = f_{(\varphi(\mu))} = f_{(\varphi(\mu'))} = f_{\{\mu'\}}$。

($\Longleftarrow$) 考虑 $\forall f\colon\ X \to X$、$f_{\{\mu\}} = f_{\{\mu'\}}$。为导出矛盾，假设 $\varphi(\mu) \neq \varphi(\mu')$。为便于阅读，我们将自动机 $i$ 首次以更新模式 $\mu$ 更新时所处的子步记为 $t_{\mu,\,i}$。于是，存在一对自动机 $(i, j)$，使得 $t_{\mu,\,i} \leq t_{\mu,\,j}$，但 $t_{\mu',\,i} > t_{\mu',\,j}$。令 $f\colon\ \mathbb{B}^n \to \mathbb{B}^n$ 为一个布尔自动机网络，使得 $f(x)_i = x_i \vee x_j$ 且 $f(x)j = x_i$，并令 $x \in \mathbb{B}^n$ 使得 $x_i = 0$ 且 $x_j = 1$。我们将比较 $f_{\{\mu\}}(x)_i$ 和 $f_{\{\mu'\}}(x)_i$，以证明矛盾。将 $f_{\{\mu\}}$ 应用于 $x$。在步 $t_\mu$ 之前，自动机 $i$ 的值仍为 0；更重要的是，由于 $t_{\mu,\,i} \leq t_{\mu,\,j}$，$j$ 的值仍为 1。这意味着紧接步 $t_\mu$ 之后，$_i$，自动机 $i$ 的值为 1，之后也不会改变。因此，我们有 $f_{\{\mu\}}(x)_i = 1$。现在将 $f_{\{\mu'\}}$ 应用于 $x$。这一次，$t_{\mu',\,i} > t_{\mu',\,j}$，这意味着自动机 $j$ 先被更新，并取此时自动机 $i$ 的值；由于自动机 $i$ 尚未更新，该值为 0。之后，两个自动机的值都不会改变，因为 $0 \vee 0$ 仍为 0。这意味着$f_{\{\mu'\}}(x)_i = 0$。因此，我们得到 $f_{\{\mu\}} \neq f_{\{\mu'\}}$，这与先前的假设相矛盾。

□

令 $\mathsf{BP}_n^0 = \mathsf{BP}_n / \equiv_0$ 表示相应的商集，即在一般情况下，为对所有可能的大小为 $n$ 的 AN 的动力学进行计算机分析而生成的块并行更新模式集合。

**定理 2.** 对于任意 $n \geq 1$，我们有：

$$|\mathsf{BP}_n^0| = \sum_{i=1}^{p(n)} \frac{n!}{\prod_{j=1}^{d(i)} (m(i,j)!)^j} \tag{1}$$

$$= \sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \prod_{\ell=1}^{j} \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,k) - (\ell-1) \cdot m(i,j)}{m(i,j)} \tag{2}$$

$$= \sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \left( \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,k)}{j \cdot m(i,j)} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot m(i,j)}{m(i,j)} \right). \tag{3}$$

证明。 $|\mathrm{BP}_n^0|$ 可以看作三个不同的公式。我们先证明这些公式相等，然后证明它们确实统计了 $|\mathrm{BP}_n^0|$。

公式 1 对于 $n$ 的每个划分（对 $i$ 求和），将所有矩阵的填充方式（$n!$）按每一列内部的置换进行合并（$m(i,\ j)!$ 对$M_j$的 $j$ 列中的每一列均如此）。

公式 2 对于 $n$ 的每个划分（对 $i$ 求和），将每个矩阵的每一列的乘积（对 $j$ 和 $\ell$ 求积）与选择元素（从剩余元素中）填充该列的方式数相乘，其中不考虑元素在列内的顺序。

公式 3 对于 $n$ 的每个划分（对 $i$ 求和），将每个矩阵的乘积（对 $j$ 求积）与选择元素（从剩余元素中）填充该矩阵的方式数相乘，再乘以使用这些元素填充矩阵各列的方式数（对 $\ell$ 求积），其中不考虑元素在每一列内的顺序。

公式 1 与公式 2 之间的等式可通过如下方式展开二项式系数得到：$\binom{x}{y} = \frac{x!}{y!\,(x-y)!}$，并注意到 $x!\ (x-y)!$ 的乘积可以约去。事实上，记 $a(j,\ \ell) = \left(n - \sum_{k=1}^{j-1} k \cdot m(i,\ k) - \ell \cdot m(i,\ j)\right)!$，则有

$$\prod_{j=1}^{d(i)} \prod_{\ell=1}^{j} \frac{(n - \sum_{k=1}^{j-1} k \cdot m(i,k) - (\ell-1) \cdot m(i,j))!}{(n - \sum_{k=1}^{j-1} k \cdot m(i,k) - \ell \cdot m(i,j))!} = \prod_{j=1}^{d(i)} \prod_{\ell=1}^{j} \frac{a(j,\ell-1)}{a(j,\ell)} = \frac{n!}{0!} = n!$$

因为 $a(1,0) = n!$，那么 $a(1,j) = a(2,0),\ \ a(2,j) = a(3,0),$ .....直到 $a(d(i)\ j) = 0!$。

公式 2 和公式 3 之间的等式是通过反复使用恒等式$\binom{x}{z}\binom{x-z}{y} = \binom{x}{z+y}\binom{z+y}{y}$得到的，由此对 $j$ 进行归纳：

$$\prod_{\ell=1}^{j} \binom{x - (\ell-1) \cdot y}{y} = \binom{x}{j \cdot y} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot y}{y}. \tag{4}$$

事实上，$j = 1$ 是显然的，并且利用关于 $j$ 的归纳假设，然后根据恒等式可得：

$$\begin{aligned}
\prod_{\ell=1}^{j+1} \binom{x - (\ell-1) \cdot y}{y} &= \binom{x - j \cdot y}{y} \cdot \prod_{\ell=1}^{j} \binom{x - (\ell-1) \cdot y}{y} \\
&= \binom{x - j \cdot y}{y} \cdot \binom{x}{j \cdot y} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot y}{y} \\
&= \binom{x}{(j+1) \cdot y} \cdot \binom{(j+1) \cdot y}{y} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot y}{y} \\
&= \binom{x}{(j+1) \cdot y} \cdot \prod_{\ell=0}^{j} \binom{(j-\ell+1) \cdot y}{y} \\
&= \binom{x}{(j+1) \cdot y} \cdot \prod_{\ell=1}^{j+1} \binom{(j+1-\ell+1) \cdot y}{y}.
\end{aligned}$$

因此，对每个 $j$ with$x = n - \sum_{k=1}^{j-1} k \cdot m(i,\ k)$ 和 $y = m(i,\ j)$ 应用方程 4，即可由公式 2 得到公式 3。

为了证明它们计算的是 $\mathrm{BP}_n^0$ 中元素的数量，我们现在论证：对于任意一对 $\mu,\ \mu' \in \mathrm{BP}_n$，$\mu \equiv_0 \mu'$ 成立，当且仅当它们的矩阵表示在列内元素的置换下相同（等价类的数量由公式 1 计算）。

在 $\varphi$ 的定义中，每个块都是一个集合，由从每个 o-块中各取一个元素构成。鉴于 $n_k$ 在 $\varphi$ 的定义中对应于定理陈述中的 $j$，一个矩阵对应于所有大小相同为 $n_k$ 的 o-块。因此，$\varphi$ 定义中的 $\ell \bmod n_k$ 运算相当于考察这些 o-块中在其矩阵表示里位于同一列的元素。由于块没有顺序之分，结论得证。

□

数列 $(|\mathsf{BP}_n^0|)_{n\geq 1}$ 的前十项为：

$$1, 3, 13, 67, 471, 3591, 33573, 329043, 3919387, 47827093.$$

它们与 OEIS [17], 中的数列 A182666 相吻合，接下来的引理将证明它们确实是同一个数列（该数列在 OEIS 中由其指数生成函数定义）。数列的指数生成函数$(a_n)_{n\in\mathbb{N}}$为 $f(x) = \sum_{n\geq 0} a_n \frac{x^n}{n!}$。

**引理 3.** ……的指数生成函数 $(|\mathsf{BP}_n^0|)_{n\in\mathbb{N}}$ 为 $\prod_{j\geq 1}\sum_{k\geq 0}\left(\frac{x^k}{k!}\right)^j$。

证明。我们将从指数生成函数入手，求出$x^n$ 的系数，并证明它等于 $\frac{|\mathsf{BP}_n^0|}{n!}$，从而证明相应的数列为$(|\mathsf{BP}_n^0|)_{n\in\mathbb{N}}$。

$$\begin{aligned}\prod_{j\geq 1}\sum_{k\geq 0}\left(\frac{x^k}{k!}\right)^j &= \left(\sum_{k\geq 0}\frac{x^k}{k!}\right)\times\left(\sum_{k\geq 0}\frac{x^{2k}}{(k!)^2}\right)\times\left(\sum_{k\geq 0}\frac{x^{3k}}{(k!)^3}\right)\times\cdots\\ &= \underbrace{\left(1+x+\frac{x^2}{2!}+\cdots\right)}_{j=1}\times\underbrace{\left(1+x^2+\frac{x^4}{(2!)^2}+\cdots\right)}_{j=2}\times\underbrace{\left(1+x^3+\frac{x^6}{(2!)^3}+\cdots\right)}_{j=3}\times\cdots.\end{aligned}$$

分布求和中的每一项都是通过为每个 $j\in\mathbb{N}_+$关联一个 $k\in\mathbb{N}$，并计算 $\frac{1}{(k!)_j}\cdot x^{jk}$ 的乘积得到的。因此，如果 $\mathbb{N}^{\mathbb{N}_+}$是从 $\mathbb{N}_+$ 到 $\mathbb{N}$ 的映射集合，我们有：

$$\prod_{j\geq 1}\sum_{k\geq 0}\left(\frac{x^k}{k!}\right)^j = \sum_{m\in\mathbb{N}^{\mathbb{N}_+}}\left(\prod_{j\geq 1}\frac{1}{(m(j)!)^j}\right)\cdot x^{\sum_{j\geq 1} j\cdot m(j)}.$$

由此，要得到 $x^n$ 的系数，我们只需对满足 $\sum_{j\geq 1} j\cdot m(j) = n$ 的映射 $m$ 求和；而这些映射恰好就是 $n$ 的分拆，其中 $m(j)$ 表示 $j$ 在该分拆中的重数。因此，$x^n$ 的系数为

$$\sum_{i=1}^{p(n)}\prod_{j\geq 1}\frac{1}{(m(i,j)!)^j} = \sum_{i=1}^{p(n)}\frac{1}{\prod_{j\geq 1}^{d(i)}(m(i,j)!)^j} = \frac{|\mathsf{BP}_n^0|}{n!}.$$

□

算法 2 枚举 $\mathsf{BP}_n^0$ 的元素。它的开头与算法 1 相同，但从第 10 行开始有所不同，即在选定 $M_j$ 的内容之后。在前一个算法中，我们需要枚举每个矩阵，且行的置换视为等价。这一次，我们需要枚举每个矩阵，且列内的置换视为等价。这意味着我们只需选择每一列的内容，而不必确定顺序。为此，我们使用另外两个函数递归枚举每列可能的内容，就像枚举矩阵的内容一样。第一个函数（第 14–20 行）以类似于函数 EnumBPeq0 的方式运行，只是不枚举划分；它主要是为了展现……

**算法 2：** $\mathrm{BP}_n^0$的枚举

```
1  函数EnumBPeq0(n):
2      对每个 i ∈分拆(n) 执行
3          EnumBPeq0aux(n,i,1)

4  函数 EnumBPeq0aux(n, i, j, M_1, ..., M_{j-1}):
5      如果 d(i) < j 则
6          枚举(M)
7          返回
8      如果 m(i,j) > 0 则
9          对组合 A 中大小为 j·m(i,j) 的每个元素执行
10             对 M_j 由 EnumBlockeq0(A, j, m(i,j)) 执行
11                 EnumBPeq0aux(n, i, j+1, M_1..... M_{j-1}, M_j)
12     else
13         EnumBPeq0aux(n, i, j+1, M_1..... M_{j-1}, ∅)

14 函数EnumBlockeq0(A, j, m):
15     对每个 C 由 EnumBlockeq0Aux(A, m)枚举执行
16         对于 k ← 1 到 m 执行
17             对于 ℓ ← 1 到 j 执行
18                 M_j[k][ℓ] = C_ℓ[k]
19         枚举(M_j)
20         返回

21 函数 EnumBlockeq0Aux(A, m, C_1, ....., C_ℓ):
22     如果 A = ∅ 则
23         枚举(C)
24         返回
25     else
26         对 A中的大小为 m 的所有组合 B 执行
27             EnumBlockeq0Aux(A \ B, m, C_1, ..., C_ℓ, B)
```

将辅助函数给出的列重塑为一个矩阵。辅助函数要么在矩阵中的每个元素都已放入某一列后返回这些列（第 22–24 行），要么枚举填充当前列的所有可能方式，并递归调用自身，以所有可能的方式填充下一列（第 25–27 行）。

**定理 3.** 算法 *2* 正确：它枚举了$\mathrm{BP}_n^0$。

证明。 首先，它为 $n$ 枚举了正确数量的更新模式。确实，对于 $n$ 的每个划分，以及从 1 到 $d(i)$ 的每个 $j$，我们先选择 $j \cdot m(i,j)$ 个元素来填充矩阵 $M_j$。然后，对于从 1 到 $j$ 的每个 $\ell$，我们选择 $\ell$ 列中的 $m(i,j)$ 个元素。因此，该算法枚举的更新模式数量如下：

$$\sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \left( \binom{n - \sum_{k=1}^{j-1} k \cdot m(i,k)}{j \cdot m(i,j)} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot m(i,j)}{m(i,j)} \right),$$

这恰好是 $\mathrm{BP}_n^0$ (定理 2) 的大小。

其次，该算法仅选择每个矩阵列的内容，而不选择其顺序。这意味着，通过该算法枚举的两种更新模式，不可能使它们各自矩阵的每一列都具有完全相同的内容，因此根据定义 1，它们不可能等价。这意味着，算法 2 能够枚举出所有在动力学相等意义下互不等价的块并行更新模式。 □

### 3.4 极限集上按动力学同构划分的序

下述定义在块并行更新模式上的等价关系，恰好刻画了具有同构极限动力学这一概念。它类似于 $\equiv_0$，区别在于可对块序列施加阶数为 $i$ 的循环移位。

**定义 2.** 对于 $\mu, \mu' \in \mathsf{BP}_n$，当存在某个$i \in [\![|\varphi(\mu')|]\!]$ （称为移位），使得 $\varphi(\mu) = \sigma^i(\varphi(\mu'))$ 时，我们记作 $\mu \equiv_\star \mu'$ 。
**注 3.** 注意， $\mu \equiv_0 \mu'$ 对应于 $\equiv_\star$的特殊情形 $i = 0$ 。因此，$\mu \equiv_0 \mu' \implies \mu \equiv_\star \mu'$。

**记号 1.** 给定 $f_{\{\mu\}}: \ X \to X$，令$\Omega_{f_{\{\mu\}}} = \bigcap_{t\in\mathbb{N}} f_{\{\mu\}}(X)$ 表示其极限集 （将 $f_{\{\mu\}}$ 的记号滥用于构型集合），并令$f^{\Omega}_{\{\mu\}} : \Omega^{f_{\{\mu\}}} \overset{\to}{} \Omega^{f_{\{\mu\}}}$ 表示其在极限集上的限制。注意，由于动力学是确定性的， $f^{\Omega}_{\{\mu\}}$ 是双射。

下面的引理表明，如果通常关注 AN 在块并行更新下的极限行为，那么研究关系 $\equiv_\star$ 的每个等价类中的一个代表元，对于获得所有可能的极限动力学谱而言，既是必要的，也是充分的。

**引理 4.** 对于任意 $\mu, \mu' \in \mathsf{BP}_n$，有 $\mu \equiv_\star \mu' \iff \forall f: \ X \to X, f^{\Omega}_{\{\mu\}} \sim f_{\Omega\{\mu'\}}$。

证明。 设 $\mu$ 和 $\mu'$ 是 $\mathsf{BP}_n$ 的两种块并行更新模式。
($\Longrightarrow$) 设 $\mu, \mu'$ 满足 $\mu \equiv_\star \mu'$，其移位为 $\hat{\imath} \in [\![p]\!]$，且 $\varphi(\mu) = (W_\ell)^{\ell\in[\![p]\!]}$, $\varphi(\mu') = (W'_\ell)^{\ell\in[\![p]\!]}$，并且 $p = |\varphi(\mu)| = |\varphi(\mu')|$。这意味着 $\forall i \in [\![p]\!]$，我们有 $W'_i = W_{i+\hat{\imath} \bmod p}$，并且对于任意自动机网络 $f$，可推得 $\pi = f_{(W_0,\dots,W_{\hat{\imath}-1})}$ 是从 $\Omega_{f_{\{\mu\}}}$ 到 $\Omega^{f_{\{\mu'\}}}$ 的所需同构。事实上， $f_{\{\mu\}}(x) = y$ 当且仅当 $f_{\{\mu'\}}(\pi(x)) = \pi(y)$，因为

$$f_{\{\mu'\}} \circ \pi = f_{(W_0,\dots,W_{\hat{\imath}-1},W'_0,\dots,W'_p)} = f_{(W'_{p-\hat{\imath}},\dots,W'_p)} \circ f_{\{\mu\}} = \pi \circ f_{\{\mu\}}.$$

注意， $\pi^{-1} = f^{(q-1)}_{\{\mu'\}} \circ f_{(W'^{\hat{\imath}}\cdots W'_{p-1})}$，其中 $q$ 为极限环长度的最小公倍数，并且 $\pi^{-1} \circ \pi$ （分别 $\pi \circ \pi^{-1}$）是 $\Omega^{f_{\{\mu\}}}$ （分别$\Omega^{f_{\{\mu'\}}}$）上的恒等映射。

($\Longleftarrow$) 我们根据 $\mu \not\equiv_\star \mu'$，通过分类讨论证明逆否命题。
（1）如果在 $\varphi(\mu)$ 和 $\varphi(\mu')$ 中，存在一个自动机 $\hat{\imath}$，其在 $\mu$ 和 $\mu'$ 中分别被更新的次数 $\alpha$ 和 $\alpha'$ 不相同，那么不失一般性，我们假设 $\alpha > \alpha'$，并考虑满足以下条件的自动机网络 $f$：

- $X_{\hat{\imath}} = [\![\alpha]\!]$ 以及对于所有 $i \neq \hat{\imath}$；并
- $f_{\hat{\imath}}(x) = (x_{\hat{\imath}} + 1)$ 模 $\alpha$ 且 $f_i(x) = x_i$ 对所有 $i \neq \hat{\imath}$。

由此可知， $f^{\Omega}_{\{\mu\}}$ 只有不动点，因为 $+1$ 模 $\alpha$ 应用了 $\alpha$ 次，而$f_{\Omega\{\mu'\}}$没有不动点，因为 $\alpha' < \alpha$。我们得出结论， $f_{\Omega_{\{\mu\}}} \not\sim f^{\Omega}_{\{\mu'\}}$。
(2) 如果在 $\varphi(\mu)$ 和 $\varphi(\mu')$ 中，所有自动机都以相同次数更新，那么从 $\mu$ 到 $\mu'$ 的变换是 $[\![n]\!]$ 上的一个置换，并且保持其矩阵表示的矩阵不变（也就是说，任意一个 $i \in [\![n]\!]$ 在 $\mu$ 和 $\mu'$中都属于大小相同的 o-块，这也意味着 $\mu$ 和 $\mu'$ 由 $n$的同一划分构造）。接下来我们考虑子情况。

(2.1) 如果 $\mu'$ 的一个矩阵无法通过对 $\mu$的列进行置换得到，那么存在一对自动机 $\hat{\imath}$, $\hat{\jmath}$，它们出现在某个$k$对应的 $\varphi(\mu)$ 的第 $k$ 个块中，但不会出现在 $\varphi(\mu')$ 的任何块中。事实上，可以取 $\hat{\imath}$, $\hat{\jmath}$，使它们在 $\mu$ 中位于同一列，但在 $\mu'$ 中位于不同列。令 $S$ 为 $\hat{\imath}$ 的 o- 块， $S'$ 为 $\hat{\jmath}$ 的 o- 块。令 $p$ 表示 $\mu$ 和 $\mu'$ 中 o- 块大小的最小公倍数。在这种情况下，我们考虑如下自动机网络 $f$：

- $X_{\hat{\imath}} = \mathbb{B} \times [\![ {}^{p}_{|S|} ]\!]$、 $X_{\hat{\jmath}} = \mathbb{B} \times [\![ {}^{p}_{|S'|} ]\!]$，以及 $X_i = \{0\}$ 对所有 $i \notin \{\hat{\imath}, \hat{\jmath}\}$。给定$x \in X$，我们将 $x_{\hat{\imath}} = (x^b_{\hat{\imath}}, x^{\ell}_{\hat{\imath}})$ 表示为 $\hat{\imath}$ 的状态（$\hat{\jmath}$ 的情况类似）；并且
- $f_{\hat{\imath}}(x) = \begin{cases} (x^b_{\hat{\jmath}}, x^{\ell}_{\hat{\imath}} + 1 \bmod {}^{p}_{|S|}) & \text{如果 } x^{\ell}_{\hat{\imath}} = 0 \\ (x^b_{\hat{\imath}}, x^{\ell}_{\hat{\imath}} + 1 \bmod {}^{p}_{|S|}) & \text{否则，} \end{cases}$ $f_{\hat{\jmath}}(x) = \begin{cases} (x^b_{\hat{\imath}}, x^{\ell}_{\hat{\jmath}} + 1 \bmod {}^{p}_{|S'|}) & \text{如果 } x^{\ell}_{\hat{\jmath}} = 0 \\ (x^b_{\hat{\jmath}}, x^{\ell}_{\hat{\jmath}} + 1 \bmod {}^{p}_{|S'|}) & \text{否则，并且} f_i(x) = \overline{x_i} \text{ 对所有 } i \notin \{\hat{\imath}, \hat{\jmath}\}。 \end{cases}$

注意，在 $\mu$ 和 $\mu'$中，$\hat{\imath}$ （分别为 $\hat{\jmath}$）在一个步内会更新 ${}^{p}_{|S|}$ （分别为 ${}^{p}_{|S'|}$）次。因此，对于任意 $x \in X$，其在 $\mu$ 和 $\mu'$ 下的两个像满足 $f_{\{\mu\}}(x)^{\ell}_{\hat{\imath}} = f_{\{\mu'\}}(x)^{\ell}_{\hat{\imath}} = x^{\ell}_{\hat{\imath}}$ （$\hat{\jmath}$的情形也类似）。因此，在一个步内 $\hat{\imath}$ 和 $\hat{\jmath}$ 的状态演化过程中，第二个元素保持不变，只有第一个元素（在 $\mathbb{B}$中）可能发生变化。我们将 $X$ 分为 $X^{=} = \{x \in X \mid x^b_{\hat{\imath}} = x^b_{\hat{\jmath}}\}$ 和$X^{\neq} = \{x \in X \mid x^b_{\hat{\imath}} \neq x^b_{\hat{\jmath}}\}$，并根据 $f_{\hat{\imath}}$和 $f_{\hat{\jmath}}$的定义，观察到以下事实：

- 在 $\mu$ 和 $\mu'$下， $X^{=}$ 的所有元素都是不动点（实际上，只有 $x^b_{\hat{\imath}}$和 $x^b_{\hat{\jmath}}$可能通过复制另一个元素而演化）。• 在 $\mu$下，令 $m$、$m'$ 分别表示在它们同步更新的 $\varphi$（$\mu$）的第 $k$个块之前，$\hat{\imath}$、$\hat{\jmath}$已更新的次数。考虑构型 $x$、$y \in X^{\neq}$，其中$x_{\hat{\imath}} = (0, -m \bmod {}^{p}_{|S|})$，$x_{\hat{\jmath}} = (1, -m' \bmod {}^{p}_{|S'|})$，$y_{\hat{\imath}} = (1, -m \bmod {}^{p}_{|S|})$以及$y_{\hat{\jmath}} = (0, -m' \bmod {}^{p}_{|S'|})$。有$f_{\{\mu\}}(x) = y$ 且 $f_{\{\mu\}}(y) = x$，因为当 $x^{\ell}_{\hat{\imath}} = x^{\ell}_{\hat{\jmath}} = 0$ 时，$x^b_{\hat{\imath}}$ 和 $x^b_{\hat{\jmath}}$ 在 $\varphi$（$\mu$）的第 $k$个块中同步交换；由于模的选取，它们在该步中不会再次交换。因此， $f\Omega_{\{\mu\}}$ 有一个长度为二的极限环。

- 在 $\mu'$条件下，对于任意 $x \in X^{\neq}$，都存在一个包含 $x^{\ell}_{\hat{\imath}} = 0$ 的子步，也存在一个包含 $x^{\ell}_{\hat{\jmath}} = 0$的子步，但它们不是同一个子步（因为在 $\mu'$中，$\hat{\imath}$和$\hat{\jmath}$从不同步）。因此， $x^b_{\hat{\imath}}$ 和 $x^b_{\hat{\jmath}}$ 最终将取得相同的值（先更新者从后更新者那里复制比特，随后后更新者复制自身的比特），即 $f_{\{\mu'\}}(x) \in X^{=}$，因此 $f\Omega_{\{\mu'\}}$ 只有不动点。

在这种情况下，我们得出 $f\Omega_{\{\mu\}} \not\sim f^{\Omega}_{\{\mu'\}}$，因为其中一个具有长度为二的极限环，而另一个只有不动点。

（2.2）如果该置换保持矩阵中的列不变（也就是说， $\mu$ 中的自动机与 $\mu'$中的自动机位于同一列），那么我们考虑最后两个子情形：

(2.2.1) 此外，如果某个矩阵的置换不是循环的（即存在三列，它们在 $\mu$ 和$\mu'$中的相对顺序并不相同），那么同一矩阵中存在三个自动机 $\hat{\imath}$、$\hat{\jmath}$ 和 $\hat{k}$，使得在 $\mu$ 中，自动机 $\hat{\imath}$ 首先更新，然后是 $\hat{\jmath}$，最后是 $\hat{k}$；而在 $\mu'$ 中，自动机 $\hat{\jmath}$ 首先更新，然后是 $k$，最后是 $\hat{\jmath}$。考虑满足以下条件的自动机网络 $f$：

- $X = \mathbb{B}^n$;
- $f_i(x) = x_{\hat{k}}$、 $f_j(x) = x_{\hat{i}}$ 以及 $f_{\hat{k}}(x) = x_{\hat{j}}$; 以及 $\hat{}$
- $f_i(x) = x_i$ 当 $i \notin \{\hat{i}, \hat{}_j, k\}$ 时。

$\hat{}$ 如果这三个自动机按 $\hat{i}$，再按 $\hat{}_j$，再按 $k$ 的顺序更新，就像 $\mu$中的情况一样，那么每次更新后，它们都会取相同的值。这意味着 $f_{\{\mu\}}$ 只有不动点，恰好是集合 $P = \{x \in \mathbb{B}^n \mid x_{\hat{i}} = x_{\hat{j}} = x_{\hat{k}}\}$。

$\hat{}$ 如果它们按 $\hat{i}$，再按 $k$，再按 $\hat{}_j$ 的顺序更新，就像 $\mu'$ 中的情况一样，不过情形要复杂一些。我们根据它们在一个周期内被更新的次数，考虑两种情况（回想一下，由于它们属于同一个矩阵，因此在子步骤中会以相同的顺序反复更新）：

- 如果它们各自被更新奇数次，那么自动机 $\hat{i}$ 和$\hat{}\,\hat{}\,\hat{}_j$ 将取自动机 $k$的初始值，而自动机 $k$ 将取自动机 $\hat{}_j$的初始值。在这种情况下， $f_{\Omega\{\mu'\}}$ 具有不动点 $P$ 和长度为二的极限环。
- 如果它们每个都更新了偶数次，则情况恰好相反：自动机 $\hat{i}$ 和 $\hat{}_j$ 将取自动机 $\hat{}_j$的初始值，而自动机 $\hat{}k$ 将保持其初始值。在这种情况下， $f_{\Omega\{\mu'\}}$ 的不动点为 $Q = \{x \in \mathbb{B}^n \mid x_i = x_j\}$，且严格包含 $P$ (即 $P \subseteq Q$和 $Q \setminus P \neq \emptyset$)。

在这两种情况下， $f^{\Omega}_{\{\mu'\}}$ 的极限集中不动点多于 $P$，因此我们得出结论：$f^{\Omega}_{\{\mu\}} \not\sim f^{\Omega}_{\{\mu'\}}$。

(2.2.2) 此外，如果所有矩阵的置换都是循环的，那么我们首先注意到，当 $\varphi(\mu)$ 和 $\varphi(\mu')$ 有一个共同的块时，它们就拥有所有共同的块（这是由置换的循环性质决定的），即 $\mu \equiv_\star \mu'$。因此，在我们的假设下，可以推导出 $\varphi(\mu)$ 和 $\varphi(\mu')$ 没有任何共同的块。由此可知，存在自动机$\hat{i}$, $\hat{}_j$，它们具有情形 (2.1) 中所述的性质，即在 $\varphi(\mu)$ 的一个块中同步，但在 $\varphi(\mu')$ 的任何块中都不会同步，而相同的构造也完成了这一证明。

□

令 $\mathrm{BP}^\star_n = \mathrm{BP}_n / \equiv_\star$ 表示相应的商集。

**定理4**。 设最小公倍数$(i)$ = 最小公倍数$(\{j \in [\![1, d(i)]\!] \mid m(i, j) \geq 1\})$。对于任意 $n \geq 1$，我们有 ：

$$|\mathrm{BP}^\star_n| = \sum_{i=1}^{p(n)} \frac{n!}{\prod_{j=1}^{d(i)} (m(i,j)!)^j} \cdot \frac{1}{lcm(i)}.$$

证明。 设 $\mu$, $\mu' \in \mathrm{BP}_n$ 是两种更新模式，使得 $\mu \equiv \mu'$。那么它们的序列形式长度相同，并且每个自动机在两者中出现的次数相同。这意味着，如果某个自动机位于 $\mu$ 的分块顺序形式中的一个大小为 $k$ 的 o-块内，那么它也位于 $\mu'$ 中一个大小相同的 o-块内。由此可知，根据定义 2 所定义的等价关系，大小为 $n$ 的两种更新模式只有在由 $n$ 的同一个划分生成时才等价。

设 $\mu \in \mathrm{BP}^0_n$ 由划分 $i$ 生成。则 $\varphi(\mu)$ 的长度为 lcm$(i)$。由于 $\mathrm{BP}^0_n$ 中不存在两个元素具有相同的块顺序形式，$\mathrm{BP}^0_n$ 中 $\mu$ 的等价类恰好包含 lcm$(i)$ 个元素，它们都由同一个划分 $i$ 生成（ $\varphi(\mu)$ 的所有块均不相同）。因此，$\mathrm{BP}^\star_n$ 中由一个划分生成的元素数量

$i$ 等于 $\mathrm{BP}_n^0$ 中由划分 $i$ 生成的元素数量，除以 $\mathrm{BP}_n^\star$ 中其等价类的元素数量，即 lcm($i$)。□

**备注 4。** $|\mathrm{BP}_n^\star|$ 的公式实际上可以通过将定理 2 中的任一公式乘以[1] 最小公倍数($i$) 并放入关于分拆的求和式中（从 $i=1$ 到 $p(n)$）得到。

虽然计算 $\mathrm{BP}_n^\star$ 的元素相当直接，但要通过确保任意两个分区序的块顺序重写在循环置换下都不相同（定义 2）来枚举它们，则更具挑战性。这由算法 3 完成。它与算法 2 非常相似，但有以下差异。在第一个函数 EnumBPiso 中：紧接着选择划分后，在一个通过归纳方式计算 lcm($i$) 的修改版算法中确定系数列表 $a[j]$ （第 3–10 行）。这些系数用于第二个辅助函数 EnumBlockIsoAux，其中强制要求矩阵 $M_j$ 的最小值 $min_j$ 位于该矩阵的前 $a[j]$ 列中（第 35–37 行；当 $min_j$ 尚未在前 $a[j]-1$ 列中被选取时，该条件得到满足；随后将其放入该列，因此只选择 $m(i,\ j)-1$ 个元素）。算法 3 的正确性将在以下陈述中论证。

**定理 5。** 算法 *3* 是正确的：它枚举 $\mathrm{BP}_n^\star$。

证明。 我们首先论证算法 3 枚举了正确数量的块并行更新模式，然后论证枚举出的任意一对 $\mu,\ \mu'$ 都满足 $\mu \not\equiv_\star \mu'$。

为便于阅读本证明的其余部分，我们将 $a[j]$ 记为 $a_j$，并将 $m(i,j)$ 记为 $m_{ij}$。根据上文所述的 $min_j$ 的放置位置（强制其位于矩阵 $M_j$ 的前 $a_j$ 列内），与算法 2 的区别在于，我们不再有 $\prod_{\ell=1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}$ 种填充矩阵 $M_j$ 的方式，而只有以下数量的方式（回忆一下，$M_j$ 有 $j$ 列和 $m_{ij}$ 行）：

$$\sum_{k=1}^{a_j}\left(\prod_{\ell=1}^{k-1}\binom{(j-\ell+1)\cdot m_{ij}-1}{m_{ij}}\right)\cdot\binom{(j-k+1)\cdot m_{ij}-1}{m_{ij}-1}\cdot\left(\prod_{\ell=k+1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}\right).$$

实际上，上述公式对每一种列的选择求和：将 $min_j$放置在从 1 到$a_j$ 的 $k$ 列中的某一列，先乘以将若干元素放入第 1 列至 $k-1$ 列的方法数（ $\ell$上的第一个乘积），再乘以选择若干元素与 $min_j$ 一起放在 $k$ 列中的方法数（中间的二项式系数），最后乘以将其他若干元素放入剩余的 $k+1$ 至 $j$列中的方法数（ $\ell$上的第二个乘积）。现在，我们有 $\binom{(j-k+1)\cdot m_{ij}-1}{m_{ij}-1}=\frac{m_{ij}}{j-k+1}\binom{}{}$ $\cdot m_{ij}\cdot\binom{(j-k+1)\cdot m_{ij}}{m_{ij}}=\frac{1}{(j-k+1)}\cdot\binom{(j-k+1)\cdot m_{ij}}{m_{ij}}$。我们还有$\binom{(j-\ell+1)\cdot m_{ij}-1}{m_{ij}}=\frac{j-l}{j-(l-1)}$ $\cdot\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}$。这意味着，选择矩阵 $M_j$ 内容的所有可能方法数可以改写如下：

$$\begin{aligned}
&\sum_{k=1}^{a_j}\left(\frac{1}{(j-k+1)}\cdot\prod_{\ell=1}^{k-1}\frac{j-\ell}{j-(\ell-1)}\cdot\prod_{\ell=1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}\right)\\
&\qquad=\sum_{k=1}^{a_j}\left(\frac{1}{(j-k+1)}\cdot\frac{j-k+1}{j}\cdot\prod_{\ell=1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}\right)\\
&\qquad=\sum_{k=1}^{a_j}\left(\frac{1}{j}\cdot\prod_{\ell=1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}\right)\\
&\qquad=\frac{a_j}{j}\cdot\prod_{\ell=1}^{j}\binom{(j-\ell+1)\cdot m_{ij}}{m_{ij}}.
\end{aligned}$$

**算法 3：** BP 的枚举$_n^\star$

```
函数EnumBPiso(n):
    对每个 i ∈分拆(n) 执行
        a 是大小为 d(i) 的列表
        b ← 1
        对于 j ← d(i) 到 1 执行
            如果 m(i,j) > 0 则
                a[j] ← 最大公约数(b, j)
                b ← 最小公倍数(b, j)
            else
                a[j] ← j
        EnumBPisoAux(n, i, 1, a)

函数 EnumBPisoAux(n, i, j, a, M_1, ....., M_{j-1}):
    if d(i) < j then
        枚举(M)
        返回
    如果 m(i,j) > 0 则
        对 A 个大小为 j · m(i,j) 的组合，在 ⟦n⟧ \ ⋃_{k=1}^{j-1} M_k 中执行循环
            min_j ← min(A)
            对 M_j 执行枚举，枚举方式为 EnumBlockIso(A, j, m(i,j), min_j, a[j]) 循环
                EnumBPisoAux(n, i, j+1, a, M_1..... M_{j-1}, M_j)
    else
        EnumBPisoAux(n, i, j+1, a, M_1, ....., M_{j-1}, ∅)

函数EnumBlockIso(A, j, m, min_j, a_j):
    对每个 C 进行枚举，EnumBlockIsoAux(A, m, min_j, a_j) 执行
        对于 k ← 1 到 m 执行
            对于 ℓ ← 1 到 j 执行
                M_j[k][ℓ] = C_ℓ[k]
        枚举(M_j)
        返回

函数EnumBlockIsoAux(A, j, m, min_j, a_j, C_1, ..., C_ℓ):
    如果 A = ∅ 则
        枚举(C)
        返回
    else
        如果 |A| = m · (j − a_j + 1) 且 min_j ∈ A 则
            对 大小为 m − 1 的组合 B，从 A \ {min_j} 中遍历执行
                EnumBlockIsoAux(A \ (B ∪ {min_j}), m, min_j, a_j, C_1, ....., C_ℓ,
                  (B ∪ {min_j}))
        else
            对大小为 m 、取自 A 的组合 B 执行
                EnumBlockIsoAux(A \ B, m, min_j, a_j, C_1, ... , C_ℓ, B)
```

因此，总体而言，该算法枚举以下数量的更新模式：

$$\sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \left( \binom{n - \sum_{k=1}^{j-1} k \cdot m_{ik}}{j \cdot m_{ij}} \cdot \frac{a_j}{j} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot m_{ij}}{m_{ij}} \right).$$

为了证明这个数等于 $|\mathsf{BP}_n^\star|$，我们需要证明 $\prod_{j=1}^{d(i)} \frac{a_j}{j} = \frac{1}{\text{最小公倍数}(i)}$。记 $L(j) = \text{最小公倍数}(\{k \in [\![j, d(i)]\!] \mid m_{ik} > 0\})$，我们通过归纳法证明，在第 5-10 行 **for** 循环的每一步结束时，我们都有：

$$\prod_{k=j}^{d(i)} \frac{a_k}{k} = \frac{1}{L(j)},$$

因此结论成立（当 $j = 1$ 时，我们得到 $L(j) = \text{最小公倍数}(i)$）。在第一步，$j = d(i)$，且

$$\frac{a_{d(i)}}{d(i)} = \frac{\gcd(\{d(i), 1\})}{d(i)} = \frac{1}{L(j)}.$$

我们假设，对于给定的 $j$，有 $\prod_{k=j}^{d(i)} \frac{a_k}{k} = \frac{1}{L(j)}$，这就是归纳假设。对于 $j-1$，存在两种可能的情况：

- 如果 $m_{i(j-1)} = 0$，那么

$$a_{j-1} = j - 1 \text{ and } \prod_{k=j-1}^{d(i)} \frac{a_k}{k} = \frac{j-1}{(j-1) \cdot L(j)} = \frac{1}{L(j-1)}.$$

- 否则，

$$\prod_{k=j-1}^{d(i)} \frac{a_k}{k} = \frac{\gcd(\{j-1, L(j)\})}{(j-1) \cdot L(j)}.$$

并且由于 $\frac{a \cdot b}{\text{最大公约数}(\{a,b\})} = \text{lcm}(\{a, b\})$，我们有

$$\prod_{k=j-1}^{d(i)} \frac{a_k}{k} = \frac{1}{\text{lcm}(\{j-1, L(j)\})} = \frac{1}{L(j-1)}.$$

由此可知，在循环结束时，我们有 $\prod_{j=1}^{d(i)} \frac{a_j}{j} = \frac{1}{\text{lcm}(i)}$，因此该算法会枚举以下数量的更新模式（参见注释 4）：

$$\sum_{i=1}^{p(n)} \prod_{j=1}^{d(i)} \left( \binom{n - \sum_{k=1}^{j-1} k \cdot m_{ik}}{j \cdot m_{ij}} \cdot \prod_{\ell=1}^{j} \binom{(j-\ell+1) \cdot m_{ij}}{m_{ij}} \right) \cdot \frac{1}{\text{lcm}(i)} = |\mathsf{BP}_n^\star|.$$

现在需要证明，该算法不会枚举出两个等价的更新模式（按照 $\equiv_\star$ 的定义）。算法 3 在很大程度上基于算法 2，因此对于给定的输入，算法 3 的输出将是算法 2 输出的子集。算法 2 枚举 $\mathsf{BP}_n^0$，这意味着它枚举的每个更新模式通过 $\varphi$ 得到的像都不相同。这意味着，算法 3 枚举的每个块并行更新模式通过 $\varphi$ 得到的像也都不相同，并且该算法不会枚举出移位为 0 的两个等价更新模式。

现在通过反证法证明，该算法不会枚举出具有非零移位的两个等价更新模式。设 $\mu, \mu' \in \mathrm{BP}_n$ 是两个更新模式，二者均由算法 3 枚举得到，且 $\mu \equiv_\star \mu'$ 具有非零移位。那么，存在 $k \in \{1, \ldots, |\varphi(\mu')| - 1\}$，使得 $\varphi(\mu) = \sigma^k(\varphi(\mu'))$，并且 $\mu, \mu'$ 都由同一个划分 $i$ 生成，其中 $|\varphi(\mu)| = |\varphi(\mu')| =$ 最小公倍数 $(i)$。此外，对于每个 $j \in [\![d(i)]\!]$，矩阵 $M_j$ 在 $\mu$ 和 $\mu'$ 中都必须包含相同的元素 $A$（因此它们在 $\varphi(\mu)$ 和 $\varphi(\mu')$ 中都每隔 $j$ 个块重复），所以特别地，$min_j$ 在两次枚举中相同。

我们将通过归纳法证明，每个 $j \in [\![d(i)]\!]$ 满足 $m(i, j) > 0$ 整除 $k$，因此 $k = 0$（等价地，$k =$最小公倍数$(i)$），从而导致矛盾。对于基本情况$j = d(i)$，我们有 $a_{d(i)} = 1$（**for** 循环第 5—10 行的第一次迭代），因此，对任意集合 $A$ 的 EnumBlockIsoAux 调用都满足第 35 行的条件，并且 $min_{d(i)}$ 会立即被选入 $C_1$（ $M_{d(i)}$的第一列）。转换为块顺序更新模式后，这意味着 $min_{d(i)}$ 出现在由 $d(i) \cdot t$ 索引的所有块中，且 $t \in \mathbb{N}$，因此 $k$ 必须是 $d(i)$ 的倍数，以便 $\sigma^k$ 将包含 $min_{d(i)}$ 的块映射到包含 $min_{d(i)}$的块。

作为归纳假设，假设对于给定的 $j$，每个 $\ell \in [\![j, d(i)]\!]$ 满足 $m_{i\ell} > 0$整除 $k$。我们将证明，$j'$（划分 $i$ 中最大的数，即满足 $m_{ij'} > 0$），也整除 $k$。在矩阵 $M_{j'}$ 中，最小的 $min_{j'}$ 被迫出现在前$a_{j'}$ 列之内。这意味着，它同时出现在 $\varphi(\mu)$ 和 $\varphi(\mu')$ 中的块索引，分别可以写成 $j' \cdot t + b$ 和 $j' \cdot t + b'$，其中$t \in \mathbb{N}$ 以及 $b, b' \in [\![1, a_{j'}]\!]$。因此，来自 $M_{j'}$ 的一个自动机在 $\varphi(\mu)$ 中处于位置$b$，在 $\varphi(\mu')$ 中处于形如 $j' \cdot t + b'$ 的位置。由此可得，$b + k = j' \cdot t + b'$可以重写为 $k = t \cdot j' + b' - b = t \cdot j' + d$，其中 $t \in \mathbb{N}$以及 $d = b' - b \in [\![-a_{j'} + 1, a_{j'} - 1]\!]$。此外，根据归纳假设，划分 $i$ 中大于 $j'$ 的每个数都整除 $k$，因此 $k$ 是这些数的公倍数。我们由此推出，它们的最小公倍数也整除 $k$。鉴于 $a_{j'}$ 是 $j'$ 与上述最小公倍数的最大公约数（第 7—8 行），这意味着 $a_{j'}$ 同时整除 $j'$ 和 $k$，从而也整除 $d$。由于 $d$ 位于 $[\![-a_{j'} + 1, a_{j'} - 1]\!]$ 中，我们有 $d = 0$，因此 $j'$ 整除 $k$。归纳证明完成。

如果划分中的每个数 $i$ 都整除 $k$，且 $k \in [\![$是$i$的最小公倍数$]\!]$，那么 $k = 0$，从而导致矛盾。这就完成了正确性证明。 □

### 3.5 实现

以下存储库中提供了三个算法1、2和3的概念验证 Python 实现：

`https://framagit.org/leah.tapin/blockpargen`.

Software Heritage 已将其存档，永久链接如下：

https://archive.softwareheritage.org/browse/directory/f1b4d83c854a4d042db5018de86b7f41ef312a07/?origin_url=https://framagit.org/leah.tapin/blockpargen.

We 已在标准笔记本电脑上进行了数值实验，结果见图2 .

## 4 结论与展望

本文首次对 AN 设置中的块并行更新模式进行了理论研究，并着重探讨了其中一些组合特征。尤其是，除了它们的

| $n$ | $\mathrm{BP}_n$ | $\mathrm{BP}^0_n$ | $\mathrm{BP}^\star{}_n$ |
|---|---|---|---|
| 1 | 1<br>- | 1<br>- | 1<br>- |
| 2 | 3<br>- | 3<br>- | 2<br>- |
| 3 | 13<br>- | 13<br>- | 6<br>- |
| 4 | 73<br>- | 67<br>- | 24<br>- |
| 5 | 501<br>- | 471<br>- | 120<br>- |
| 6 | 4051<br>- | 3591<br>- | 795<br>- |
| 7 | 37633<br>- | 33573<br>0.103秒 | 5565<br>- |
| 8 | 394353<br>0.523秒 | 329043<br>0.996秒 | 46060<br>0.161秒 |
| 9 | 4596553<br>6.17秒 | 3919387<br>12.2秒 | 454860<br>1.51秒 |
| 10 | 58941091<br>1分钟24秒 | 47827093<br>2分钟40秒 | 4727835<br>16.3秒 |
| 11 | 824073141<br>21分钟12秒 | 663429603<br>38分钟31秒 | 54223785<br>3分钟13秒 |
| 12 | 12470162233<br>5小时27分钟38秒 | 9764977399<br>9小时49分钟26秒 | 734932121<br>45分钟09秒 |

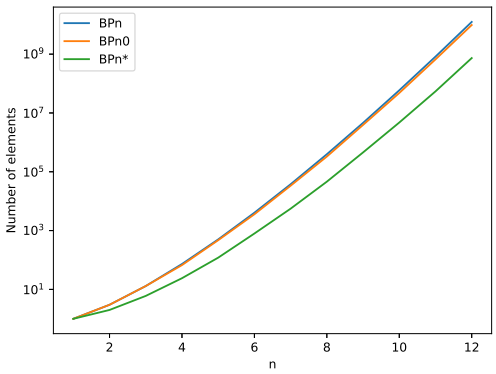


图2：我们的 Python 实现算法1、2和3在标准笔记本电脑（处理器 Intel-Core™ i7 @ 2.80 GHz）上的数值实验。对于 $n$ 从1到12，表格（左）给出了 $\mathrm{BP}_n$、$\mathrm{BP}^0_n$ 和 $\mathrm{BP}^*_n$ 的大小，以及枚举其元素所需的运行时间（每个等价类取一个代表；破折号表示运行时间小于 0.1 秒）；图表（右）则以对数尺度展示了它们各自的大小。可以看出，$\mathrm{BP}_n$ 和 $\mathrm{BP}^0_n$ 的大小相近，而 $\mathrm{BP}^*_n$则使规模提升了一个数量级，这对于研究块并行更新模式下极限动力学的高级数值实验可能具有重要意义。

从集合论角度进行一般计数和枚举，并给出澄清其与更经典的块顺序更新模式之间交集的结果后，我们给出了块并行更新模式的简洁结果，以及它们能够产生的基于AN的动力系统。事实上，通过以关于其数量的组合证明为基础，我们在两种情况下对这类模式进行了算法枚举：第一种旨在形式化地强调块并行更新模式的代表元，使其能够生成所有可能的不同底层动力系统；第二种沿用相同思路，处理那些我们可以保证不会生成具有相同极限环集合（极限动力学）的动力系统的块并行更新模式。

在枚举算法的这一背景下，最先想到的一个自然视角是考察其复杂度。首先，尽管仍需证明，算法1、2和3似乎属于 EnumP [6]。确实，这些算法的输出相对于自动机数量具有多项式规模（根据分区序的定义），且对于每个

这些算法中，判断一个子列表集合是否属于它们的输出，先验地属于 P类。然而，仍有一个问题：它们属于哪些 EnumP 子类？直观上，它们似乎应该属于 DelayP，但这个问题仍有待探讨。

块并行更新模式的特殊之处在于，它们会在一个周期内产生局部更新重复。确实，所有非块顺序的块并行更新模式都是如此（即 当更新模式定义为分区序时，至少有两个大小不同的块，参见引理 1）。既然我们知道，局部更新重复可能会破坏块顺序自动机网络中的不动点不变性（参见注1中的示例），那么，刻画这些重复与局部函数所定义的自动机之间相互作用架构的关联条件，并确定在何种条件下所得到的不动点集合会不同于并行更新模式下得到的集合，应当是很有意义的。更一般地，作为对第3.4节结果的补充，研究以下问题将颇有意义：给定一个自动机网络 $f$，其 $f$块并行敏感性/鲁棒性在多大程度上成立？在 [2, 1, 3], 中，作者通过提出更新有向图这一概念，研究了块顺序布尔自动机网络中的这一问题；该概念能够在句法层面刻画动力学等价的条件。然而，一旦涉及局部更新重复，这一概念便不再适用。因此，在周期更新模式的一般情形下创立一种新的更新有向图概念，将是解释和理解自动机网络更新敏感性/鲁棒性的重要一步。

另一个研究方向是理解基本相互作用环在采用块并行更新时如何演化。例如， [14]的作者已经表明，布尔情形下的这类环对于块并行顺序更新模式变体具有某种程度的稳健性：长度为 $p$ 的极限环数量，与在并行模式下演化的一个更小环（符号相同）的极限环数量相同。结合 [8], 的组合分析，这一结果为布尔相互作用环的渐近动力学提供了完整分析。这引出了以下问题：相互作用环的行为是否会因块并行更新模式变体而表现出类似的性质？在这里，局部更新重复同样应发挥关键作用。这项研究可以作为上述更一般问题的初步探索，因为众所周知，环是自动机网络的行为复杂性引擎 [34]。

最后，由于块并行更新模式构成了一类新的更新模式，与之相关的研究领域目前仍完全处于开放状态，我们认为，本文一个颇具前景的研究方向是沿着 [10] 关于反应系统的思路，研究自动机网络中经典判定问题的复杂度。这里需要回答的一般性问题是：由块并行更新模式诱导的局部更新重复，是否会使这类判定问题处于多项式层级中更高的层次，甚至达到多项式空间完全性?

**致谢** 作者的主要经费来源是作为法国国家公务员所领取的薪资。此外，本研究还获得了ANR-18-CE40-0002 FANs项目（KP和SS）、ECOS-Sud C19E02 SyDySy项目（SS）以及STIC AmSud 22-STIC-02 CAMA项目（KP、LT和SS）的资助。

## 参考文献

[1] J.Aracena、É.Fanchon、M.Montalva 和M.Noual。布尔网络更新有向图中的组合学。《离散应用数学》，159:401–409，2012年。

[2] J. Aracena、E. Goles、A. Moreira 和 L. Salinas。布尔网络更新调度的鲁棒性。《生物系统》，97:1–8，2009年。[3] J. Aracena、L. Gómez 和 L. Salinas。布尔网络中的极限环与更新有向图。《离散应用数学》，161:1–12，2013年。[4] E. Berlekamp、J. Conway 和 R. Guy。《数学博弈制胜之道》。学术出版社，1982年。[5] M. Cook。初等元胞自动机中的普适性。《复杂系统》，15:1–40，2004年。[6] N. Creignou、M. Kröll、R. Pichler、S. Skritek 和 H. Vollmer。困难枚举问题的复杂性理论。《离散应用数学》，268:191–209，2019年。[7] M. I. Davidich 和 S. Bornholdt。布尔网络模型预测裂殖酵母的细胞周期序列。《公共科学图书馆·综合》，3:e1672，2008年。[8] J. Demongeot、M. Noual 和 S. Sené。布尔自动机电路动力学的组合学。《离散应用数学》，160(4–5):398–415，2012年。[9] J. Demongeot 和 S. Sené。关于分块并行布尔网络：一篇立场论文。*Natural Computing*，19:5–13，2020年。[10] A. Dennunzio、E. Formenti、L. Manzoni 和 A. E. Porreca。反应系统动力学的复杂度。《信息与计算》，267:96–109，2019年。[11] B. Elspas。自主线性时序网络理论。《*IRE*电路理论汇刊》，6:45–60，1959年。[12] B. Fierz 和 M. G. Poirier。染色质动力学的生物物理学。《生物物理学年度评论》，48:321–345，2019年。[13] G. Gamard、P. Guillon、K. Perrot 和 G. Theyssier。自动机网络的Rice型定理。在*STACS'21*论文集中，第187卷*LIPIcs*，第32:1–32:17页。Schloss Dagstuhl—莱布尼茨信息学中心，2021年。[14] E. Goles 和 M. Noual。分块顺序更新调度与布尔自动机电路。在*AUTOMATA'2010*论文集中，第41–50页。DMTCS，2010年。[15] E. Goles 和 L. Salinas。布尔网络并行动力学与串行动力学的比较。*Theoretical Computer Science*，396:247–253，2008年。[16] M. R. Hübner 和 D. L. Spector。染色质动力学。《生物物理学年度评论》，39:471–489，2010年。[17] OEIS基金会。在线整数序列百科全书中的条目A182666，2023年。https://oeis.org/A182666。[18] OEIS基金会。Fubini数。在线整数序列百科全书中的条目A000670，2023年。https://oeis.org/A000670。[19] OEIS基金会。“列表集”的数量。在线整数序列百科全书中的条目A000262，2023年。https://oeis.org/A000262。

[20] OEIS基金会。将 $n$ 数量相等的带标签元素分入带标签盒子中的方法数。在线整数序列百科全书中的条目 A061095，2023年。https://oeis.org/A061095.[21] J. Kari。元胞自动机极限集的 Rice定理。*Theoretical ComputerScience*，127:229–254，1994年。[22] G. Karlebach 和 R. Shamir。基因调控网络的建模与分析。*Nature Reviews Molecular Cell Biology*，9:770–780，2008年。[23] S. A. Kauffman。随机构造的遗传网络中的代谢稳定性与表观发生。*Journal of Theoretical Biology*，22:437–467，1969年。[24] J. Kelleher。将分拆编码为升序组合。博士论文，科克大学，2005年。[25] S. C. Kleene。自动机研究，Annals of Mathematics Studies 第34卷，第3–41页，章节：神经网络与有限自动机中的事件表示。普林斯顿大学出版社，1956年。[26] W. S. McCulloch 和 W. Pitts。神经活动中内在观念的逻辑演算。*Journal of Mathematical Biophysics*，5:115–133，1943年。[27] L. Mendoza 和 E. R. Alvarez-Buylla。拟南芥 花形态发生的遗传调控网络动力学。*Journal of Theoretical Biology*，193:307–319，1998年。[28] M. Noual。自动机网络的更新。博士论文，里昂高等师范学院，2012年。[29] M. Noual 和 S. Sené。迈向布尔自动机网络建模理论——I：理论化与观察，2011年。arXiv:1111.2077。[30] M. Noual 和 S. Sené。单调布尔自动机网络中的同步与异步。*Natural Computing*，17:393–402，2018年。[31] L. Paulevé 和 S. Sené。系统生物学建模与分析：形式生物信息学方法与工具，第173–250页，章节：布尔网络及其动力学：更新的影响。Wiley，2022年。[32] M. Ríos Wilson 和 G. Theyssier。论对称性与异步：自动机网络中普适性的边界。arXiv:2105.08356，2021年。[33] F. Robert。有限集上的迭代与收缩元胞自动机。*Linear Algebra and its Applications*，29:393–412，1980年。[34] F. Robert。离散迭代：度量研究，Springer计算数学丛书第6卷。Springer，1986年。[35]I. Shmulevich、E. R. Dougherty 和 Wei Zhang。从布尔网络到作为基因调控网络模型的概率布尔网络。*Proceedings of the IEEE*，90:1778–1792，2002年。[36] A.R. Smith III。简单的计算普适元胞空间。*Journal of the ACM*，18:339–353，1971年。

[37] D. Thieffry 和 R. Thomas。生物调控网络的动力学行为——II：λ噬菌体中的免疫控制。*Bulletin of Mathematical Biology*，57:277–297，1995年。[38] R. Thomas。遗传控制电路的布尔形式化。*Journal of TheoreticalBiology*，42:563–585，1973年。[39] J. von Neumann。自复制自动机理论。伊利诺伊大学出版社，1966年。由 A. W. Burks 编辑并补全。